

On Identifying Global Optima in Cooperative Coevolution

Anthony Bucci
DEMO Lab
Computer Science Department
Brandeis University
Waltham, MA 02454
abucci@cs.brandeis.edu

Jordan B. Pollack
DEMO Lab
Computer Science Department
Brandeis University
Waltham, MA 02454
pollack@cs.brandeis.edu

ABSTRACT

When applied to optimization problems, Cooperative Coevolutionary Algorithms (CCEA) have been observed to exhibit a behavior called relative overgeneralization. Roughly, they tend to identify local optima with large basins of attraction which may or may not correspond to global optima. A question which arises is whether one can modify the algorithm to promote the discovery of global optima. We argue that a mechanism from Pareto coevolution can achieve this end. We observe that in CCEAs candidate individuals from one population are used as tests or measurements of individuals in other populations; by treating individuals as tests in this way, a finer-grained comparison can be made among candidate individuals. This finer-grained view permits an algorithm to see when two candidates are differently capable, even when one's evident value is higher than the other's. By modifying an existing CCEA to compare individuals using Pareto dominance we have produced an algorithm which reliably finds global optima. We demonstrate the algorithm on two *Maximum of Two Quadratics* problems and discuss why it works.

Categories and Subject Descriptors

G.1.6 [Mathematics of Computing]: Optimization; I.2.8 [Problem Solving, Control Methods, and Search]: Heuristic Methods

General Terms

Algorithms, Theory

Keywords

coevolution, cooperative coevolution, Pareto coevolution

1. INTRODUCTION

The originally-stated aim of Cooperative Coevolutionary Algorithms (CCEAs) was to attack the problem of evolving

complicated objects by explicitly breaking them into parts, evolving the parts separately, and then assembling the parts into a working whole [8]. The intuition is that if we know what makes a good part, then it should be easier to find good parts and assemble them than it would be to find a working whole directly. However, what makes a part good is not clear and was not directly addressed by Potter in [8].

Wiegand has established the importance of this question by pointing out that CCEAs do not optimize when they use certain straightforward notions of what constitutes a good part [11]. For instance, one might assess the value of a part by testing the value of a whole made from that part and other necessary parts. By repeating this process for a variety of combinations, one can then take an average or maximum value of the wholes produced and develop a sense for how good the part is. Wiegand has shown that when this is done, CCEAs exhibit a behavior called *relative overgeneralization*. Rather than finding complete objects which are optimal with respect to the problem, CCEAs tend to find objects which are robust under a change of parts, the *robust resting balance* [11]. In certain problems, robust resting balance objects are not globally optimal. In cases where a robust solution is preferred to a globally optimal one, relative overgeneralization is in fact a desirable behavior. However, if one wishes to optimize with a CCEA, relative overgeneralization is undesirable. The question becomes: how can we modify the CCEA to encourage it to find global optima?

Panait et al. have approached this question by biasing the assessment of a part towards what its optimal assessment should be [7]. The technique is to assess a part in the context of other coevolving parts, just as in ordinary CCEAs, but to mix this with an estimate of the optimal assessment. In the studies published in [7], the optimal assessment was given to the algorithm in advance. The authors acknowledge that such information is not available in real problem domains and offer the hope that a useful estimate of this value can be built as an algorithm runs. Whether this is the case remains to be seen.

Here we take a different perspective on the question of how to encourage a CCEA to optimize. Research on competitive coevolution [9] made clear that averaging an individual's performance against coevolving opponents was the source of several pathologies. Pareto coevolution was proposed as a remedy for these difficulties [4, 6]. The idea behind Pareto coevolutionary algorithms is to assess individuals not in terms of average performance but in terms of a list of outcomes, one for each opponent. Individuals can then be compared on an opponent-by-opponent basis

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO '05, June 25–29, 2005, Washington, DC, USA.
Copyright 2005 ACM 1-59593-010-8/05/0006 ...\$5.00.

using a comparison called *Pareto dominance* borrowed from Evolutionary Multiobjective Optimization (see, for instance [5]). It has been shown formally that this technique virtually eliminates one of the more serious pathologies, intransitivity [2] (see also the discussion of monotonic solution concepts in [3]). Pareto coevolution also highlights issues like over-specialization which are otherwise not obvious [1] (see also [10]). Can these lessons be brought to bear in cooperative coevolution?

Following Panait et al. [7], we augment a traditional CCEA with a Pareto dominance mechanism for comparing individuals. We study its performance on the same *Maximum of Two Quadratics* function studied by Panait et al. and first presented in [11]. What we observe is that the augmented algorithm reliably discovers both the global optimum of the test function. We present an analysis of this test function on the basis of the framework developed in [2] to explain why this algorithm works. These results suggest the beginnings of a fruitful cross fertilization between Pareto coevolution and cooperative coevolution research.

This paper is organized as follows. In section 2 we give necessary background from Pareto coevolution and cooperative coevolution. In section 3 we give and analyze the particular *Maximum of Two Quadratics* used in [7] and in the experiments reported here. In section 4 we present our experimental results. Finally, in section 5 we discuss implications of what we have observed.

2. BACKGROUND

In this section we will review the cooperative coevolutionary framework and give the details of the *Maximum of Two Quadratics* function and the CCEA studied in [7]. We will then briefly review the mathematical framework for Pareto coevolution laid out in [2], giving enough detail to explicate the Pareto dominance mechanism we will be adding to the CCEA and the informativeness mechanism which we will be using to analyze *MTQ* functions.

2.1 Pareto Coevolution

Much of the material we are reviewing here can be found in more detail in [2], which should be consulted for details.

Let $p : S \times T \rightarrow R$ be a function; here S and T are sets and R is an ordered set. The set S can be thought of as the set of *candidate solutions* (candidates). These are the entities which an algorithm is meant to optimize. The set T can be thought of as *tests*; these individuals serve to test or measure the candidates and give information about how good they are. The ordered set R can be thought of as *results* or outcomes of interactions between candidates and tests.

It is worth pointing out that both candidates and tests are roles that individuals can take. When we have an algorithm like a CCEA over a function $f : X \times Y \rightarrow \mathbb{R}$, we are in much the same situation as above: a two-place function into an ordered set. However, the situation is slightly different. The CCEA aims to optimize both individuals in X and individuals in Y ; thus these sets act variously as candidates and tests. At the stage of the algorithm when X elements are selected, X is acting as the candidate set to be optimized; the possible collaborators from Y can then be thought of as tests measuring the X values. However, when we consider updating Y , the reverse situation occurs: the Y values are candidates and the X values are tests.

Let us return to the function p . We can use this function to compare two candidates $s_1, s_2 \in S$ as follows: s_2 (Pareto) *covers* s_1 , which we will denote $s_1 \preceq s_2$ if, for all $t \in T$, $p(s_1, t) \leq p(s_2, t)$. If each $t \in T$ is thought of as a test, this relation says that for each test, s_2 does at least as well as s_1 . s_2 *Pareto dominates* s_1 if $s_1 \preceq s_2$ and, in addition, there is a $t \in T$ such that $p(s_1, t) < p(s_2, t)$. That is, s_2 is at least as good as s_1 against all tests and is strictly better than s_1 against at least one test.

If it happens that there are $t_1, t_2 \in T$ such that $p(s_1, t_1) < p(s_2, t_1)$ but $p(s_1, t_2) > p(s_2, t_2)$, then s_1 and s_2 are *incomparable*. We will denote incomparability $s_1 \diamond s_2$. The idea here is that there are two tests, one which shows s_2 is better than s_1 , but another which shows s_1 is better than s_2 . In this case we cannot give a strict relationship between the two candidates; they each have value, albeit in different ways.

The relation \preceq is a preorder on the set S . As such it has a set of maximal elements; these are elements $\hat{s} \in S$ such that if $\hat{s} \preceq s$, then $s \preceq \hat{s}$ must also hold, for any $s \in S$. In words, \hat{s} is maximal if, whenever it appears to be less than or equal to some other value, that value is less than or equal to it (meaning the two values are equivalent). The set of all maximal elements of \preceq is called the *Pareto optimal set*, *Pareto front*, or *non-dominated front* by various authors.

Thus far we have considered comparing candidates in S . What about the tests in T ? An observation made in [4] is that it is not appropriate to use Pareto covering or Pareto dominance to compare tests. Rather, as advocated there and in [2], it makes sense to compare tests according to how *informative* they are about ranking candidates. It suffices for our purposes to consider two cases. If two tests $t_1, t_2 \in T$ rank all the candidate individuals in the same order, then they are *equally informative*. However, if there are $s_1, s_2 \in S$ such that $p(s_1, t_1) < p(s_2, t_1)$ and $p(s_1, t_2) > p(s_2, t_2)$, then the tests t_1 and t_2 are *differently informative*.

2.2 Cooperative Coevolution

Cooperative coevolution has generally been presented as a framework as opposed to a particular algorithm. We will follow the scheme given in algorithm 2 of [11]. The algorithm keeps some indexed set of populations p_s . For each population p_s , parents are selected, offspring generated via the variation operators, collaborators are selected from the remaining populations, and the individuals of p_s are evaluated with these collaborators. The next generation for population p_s consists of the individuals which survive selection based on this evaluation. Within this framework one is free to use whichever selection, generation, variation, collaboration, and evaluation mechanism one wishes. In what follows we will detail the choices which were made for the experiments reported in section 4.

Some applications of cooperative coevolution involve a function of form $f : X \times Y \rightarrow \mathbb{R}$. The task is to find values in the sets X and Y which optimize this function. One way to approach optimizing such a function is to define two populations corresponding to settings for X and Y . The evaluation of two individuals $x \in X$ and $y \in Y$ can then be $f(x, y)$ or some function thereof. Relative overgeneralization is already evident in simple scenarios like this one.

We will be considering two *Maximum of Two Quadratics* (*MTQ*) functions [11]. Let f_1 and f_2 be two quadratic functions defined:

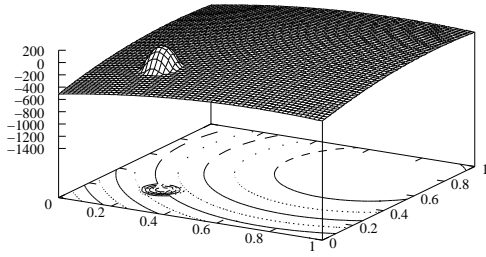


Figure 1: Plot of MTQ_1 .

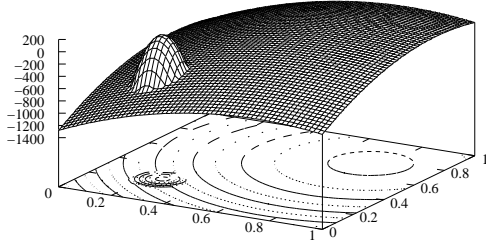


Figure 2: Plot of MTQ_2 .

$$f_1(x, y) = h_1 \left(1 - \frac{16(x - x_1)^2}{s_1} - \frac{16(y - y_1)^2}{s_1} \right)$$

$$f_2(x, y) = h_2 \left(1 - \frac{16(x - x_2)^2}{s_2} - \frac{16(y - y_2)^2}{s_2} \right)$$

where h_i , s_i , x_i and y_i are parameters controlling the height, width, and vertex of the function. Given two such functions, an MTQ function is defined:

$$MTQ(x, y) = \max(f_1(x, y), f_2(x, y)) \quad (1)$$

The function employed in [7] uses the following parameter settings: $h_1 = 50$, $s_1 = 1.6$, $(x_1, y_1) = (\frac{3}{4}, \frac{3}{4})$; and $h_2 = 150$, $s_2 = \frac{1}{32}$, $(x_2, y_2) = (\frac{1}{4}, \frac{1}{4})$.¹ We will denote this function MTQ_1 . We will also be considering a second function MTQ_2 which is MTQ_1 but with $h_1 = 125$. These two functions are displayed in Figs. 1 and 2.

In our experiments we will be comparing three CCEAs. The first is as in [7]; for brevity we will call this algorithm simply CCEA. The sets X and Y are both sets of reals, the unit interval $[0, 1]$. There are two populations, one consisting of 32 individuals from X , the other containing 32 individuals from Y . Tournament selection is used. Here, to perform selection on X for example, two individuals x_1, x_2 are selected at random from the X population. These individuals are varied by adding Gaussian noise with mean 0 and standard deviation 0.05. Then the individuals are evaluated. Evaluation is done by selecting the highest-valued individual y^* from the Y population of the previous generation² as well as a random individual y . x_1 's evaluation is $\max(MTQ(x_1, y^*), MTQ(x_1, y))$; i.e., the objective function value of x_1 and its best collaborator. Once evaluation is performed, a random value r between 0 and 1 is chosen. If $r \leq k$, then the higher-valued of x_1, x_2 is added to the next

¹We have switched the positions of the global and local optima versus what was reported in Panait et al.

²Or from the current generation at step 1 of the algorithm

population; otherwise the lower-valued individual is added. Here k is a parameter controlling the selection pressure; we have used $k = 0.75$ in all experiments reported. One last point is that the highest-valued individual of the previous generation is always carried forward unmodified to the next generation.

The second CCEA variant, which we will call pCCEA, has two modifications over CCEA. First, at the stage when individuals are being compared, objective value with a collaborator is not used. Instead, the two individuals are compared according to Pareto dominance. That is, instead of considering adding the individual with the higher objective value, we consider adding the dominant individual; for instance, if $x_1 \preceq x_2$ and $r \leq k$, then x_2 is added to the next generation. If $x_1 \diamond x_2$, both are added to the next generation. Second, since there is no notion of best individual in this case, we will carry forward the non-dominated front of the previous generation. It is possible that dominated individuals make it into the population; thus, we first find the Pareto front of the current population, carry that forward to the next generation, and fill in any remaining space in the population with individuals selected as above.

The final CCEA, which we call cCCEA, has one modification over CCEA. Instead of comparing individuals according to the best setting of the previous generation, we will have each individual collaborate with all individuals in the other population and give it the highest value it receives from all these collaborations. We will use cCCEA as a control for pCCEA: since the Pareto dominance mechanism in pCCEA has access to all individuals in the test population, it seems only fair to give CCEA the same information and see how well it performs.

3. ANALYSIS

In this section we make two observations:

1. The initial populations of 32 X and Y settings frequently represent individuals on the higher peak. That is, there is a setting in the X population and a setting in the Y population which, when paired together, lie on the higher-valued peak;
2. In the domain defined by MTQ_1 , the globally-optimal and locally-optimal settings for one population are Pareto optimal as well as being differently informative when treated as tests of the other population.

The import of the first observation is that the initial population of these algorithms already contains individuals which could potentially move to the global optimum. In the case of the CCEA without the Pareto dominance mechanism, the experimental observation that the algorithm often does not find this global optimum implies that the algorithm is actively moving *away* from the higher-valued peak.

The import of the second observation is that, unlike the situation in competitive domains, a CCEA running on these MTQ functions does not require an explicit informativeness mechanism to keep informative tests in the population. Pareto dominance suffices. The reason is that in competitive domains, Pareto dominant individuals tend to make poor tests; thus there is a need for a separate mechanism to encourage informative tests to remain in the population. In MTQ_1 at least, the situation is different: Pareto optimal

(x, y)	f_1	f_2
(0.2, 0.2)	-252.5	-225.0
(0.2, 0.29)	-207.05	-157.5
(0.29, 0.2)	-207.05	-157.5
(0.29, 0.29)	-161.6	-90.0

Table 1: Values of f_1 and f_2 on four points in the xy -plane spanning a square which lies under the higher-valued peak.

settings tend to also be informative tests of their collaborators.

3.1 Initial Populations and the Higher Peak

We will show that, with an initial population of 32 X and Y individuals, the initial population of a CCEA has roughly a 90% chance of containing a representative on the higher-valued peak. To prove this, we will show that all points (x, y) in the square spanned by the points (0.2, 0.2) and (0.29, 0.29) are such that $f_2(x, y) > f_1(x, y)$; i.e. are such that $MTQ_1(x, y) = f_2(x, y)$. As a result, the points are within this square are all on the higher-valued peak. We will then calculate the probability that the initial population contains at least one point in this square; this probability will give a lower bound on the probability that the population contains a representative on this peak.

Because the region for which this relation holds is simply connected, it suffices to show the corners of the square are all such that $f_2(x, y) > f_1(x, y)$. In table 1 we give the values of f_1 and f_2 for the four corners of this square; for all four points f_2 's value is larger.

The probability that a value chosen uniformly randomly in the range $[0, 1]$ will land in the subinterval $[0.2, 0.29]$ is $p = 0.09$. Now, if 32 values are chosen uniformly randomly, the chance that at least one of them will lie in the subinterval is $1 - (1 - p)^{32}$ or approximately 0.95. In other words, the chance that the initial population of X values has at least one individual in this range is roughly 0.95; similarly for Y . Thus, the probability that the initial population has at least one X value and one Y value in $[0.2, 0.29]$ is roughly $0.95 \cdot 0.95$ or roughly 0.90. In short, roughly 90% of runs of a CCEA with initial population of 32 X and Y values should have at least one representative on the higher-fitness peak.

3.2 Dominance and Informativeness

Much can be said about the dominance and informativeness structure in the domain defined by MTQ_1 . We will simply show that the globally- and locally-optimal individuals are Pareto optimal when treated as candidates and differently informative when treated as tests. Note that because of the symmetry of the MTQ_1 function, all statements made treating X individuals as candidates apply equally well when Y values are treated as candidates. Similarly, statements about Y individuals as tests apply when X individuals are treated as tests.

Regarding dominance, let the Y values be candidates and X values tests. Let $y^* = \frac{1}{4}$ be the globally-optimal setting for Y and let $y_* = \frac{3}{4}$ be the locally-optimal setting. Fig. 3 depicts these two candidates as functions of X . Note they are incomparable. For $x_1 \in (x, x')$, $MTQ_1(x_1, y_*) < MTQ_1(x_1, y^*)$. However, for $x_2 \notin [x, x']$, $MTQ_1(x_2, y_*) > MTQ_1(x_2, y^*)$. Thus, there is a test, x_1 which says that y^*

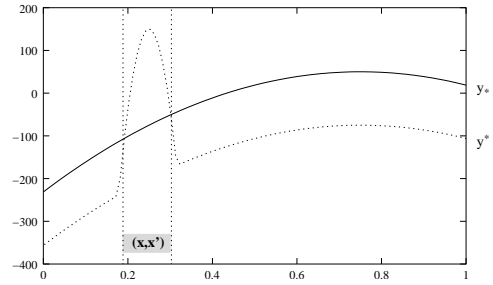


Figure 3: Plot of candidates y^* and y_* as cross sections through MTQ_1 . The shaded interval (x, x') consists of those X settings for which y^* is better than y_* ; for the remaining X settings, y_* is better than y^* , making the two non-dominated.

is better than y_* and a test x_2 which says that y_* is better than y^* .

Furthermore, both these settings are Pareto optimal. For y_* , note that the corresponding x_* is such that for all $y \in Y$, $MTQ_1(x_*, y) < MTQ_1(x_*, y_*)$. Consequently, y_* cannot be dominated and so must lie on the Pareto front. Similarly, for y^* , $MTQ_1(x^*, y) < MTQ_1(x^*, y^*)$, meaning y^* cannot be dominated. Therefore both y_* and y^* are Pareto optimal as candidates.

Now let us consider informativeness when treating Y settings as tests. Partition Y into two subsets T_1 and T_2 as follows: $T_1 = \{y \in Y \mid \forall x \in X, MTQ_1(x, y) = f_1(x, y)\}$ and $T_2 = Y \setminus T_1$. Note that $y_* \in T_1$ (because all points (x, y_*) are on the lower-valued peak; see fig 3) and $y^* \in T_2$.

First we observe that all tests in T_1 are equally informative. Recall that this means they all put the X settings in the same order. Let $y, y' \in T_1$ and define two functions on X :

$$g(x) = f_1(x, y)$$

$$h(x) = f_1(x, y')$$

Simply, g gives the value of $x \in X$ when y is applied to it; h gives the value of x when y' is applied. Now let $x, x' \in X$. Then $g(x) \leq g(x')$ if and only if $h(x) \leq h(x')$. To see this, notice that $g(x) + h_1 \frac{16(y-y_1)^2}{s_1} - h_1 \frac{16(y'-y_1)^2}{s_1} = h(x)$ for any x . Since adding a quantity to both sides of an inequality does not change its direction, $g(x) \leq g(x')$ implies $h(x) \leq h(x')$. A symmetrical argument shows that $h(x) \leq h(x')$ implies $g(x) \leq g(x')$. Therefore g and h both induce the same order on X by pullback [2]. Since $g(x) = f_1(x, y) = MTQ_1(x, y)$ and $h(x) = f_1(x, y') = MTQ_1(x, y')$, it follows that y and y' order X in the same way. In other words y and y' are equally informative tests. In particular, since $y_* \in T_1$, this test orders X in the same way as all other tests in T_1 .

The situation in T_2 is more complicated than in T_1 . It is beyond our scope to go into a detailed analysis of the informativeness structure for T_2 . However, we observe that the tests in T_2 are differently informative than those in T_1 . To see this, let $y \in T_1$ and $y' \in T_2$, let $x^* = \frac{1}{4}$ be the globally-optimal X setting and let $x \in X$ be such that:

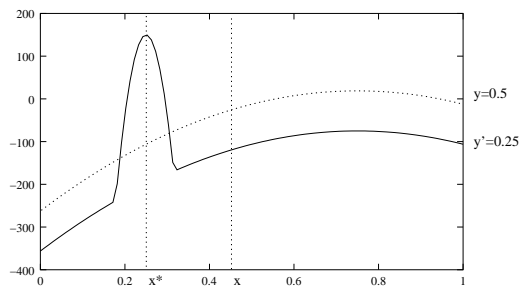


Figure 4: Cross sections through MTQ_1 defined by two different Y values. x^* and x depicted here satisfy relations 2 through 4.

$$f_1(x^*, y) < f_1(x, y) \quad (2)$$

$$f_2(x, y') < f_1(x, y') \quad (3)$$

$$f_1(x, y') < f_2(x^*, y') \quad (4)$$

Fig. 4 illustrates these relationships.

Relation 2 implies that $MTQ_1(x^*, y) < MTQ_1(x, y)$, in other words that test y ranks x^* strictly lower than x . Relation 3 implies that $MTQ_1(x, y') = f_1(x, y')$. By definition of x^* , $MTQ_1(x^*, y') = f_2(x^*, y')$. Thus, relation 4 implies that $MTQ_1(x, y') < MTQ_1(x^*, y')$ or, in other words, that test y' ranks x^* strictly higher than x . In short, the tests y and y' rank x and x^* differently, so are differently informative. y and y' were chosen arbitrarily from T_1 and T_2 , respectively, meaning that any test in T_1 is differently informative from any other test in T_2 . In particular, y_* and y^* are differently informative. Recall that these two settings are also Pareto optimal; therefore we have shown that two Pareto optimal settings for Y also make differently informative tests, indicating a close relationship between Pareto dominance and informativeness which is not typically present in competitive domains.

4. EXPERIMENTS

In this section we will report on the experiments performed. We replicate the case reported in [7] on MTQ_1 when $\delta = 0$. We then perform the same experiment, this time using Pareto dominance to compare individuals as outlined in section 2.2.

We ran CCEA, pCCEA, and cCCEA for 50 generations on MTQ_1 . We repeated this experiment 250 times for each algorithm. Table 2 reports the number of runs which found an individual near the global optimum³ and gives the mean objective value of the highest-valued individual from each of the 250 runs. Recall that the objective value of the higher peak is 150, while the objective value of the lower peak is 50. CCEA never finds high-quality individuals; instead it always finds individuals at or near the local optimum, corroborating what was observed in [7] (see, for instance, Fig. 2 in that paper for the case $\delta = 0$). By contrast, pCCEA reliably finds individuals at or near the global optimum.

However, cCCEA also tends to find individuals near the higher peak. The question arises whether pCCEA succeeds

³By which we mean the X and Y settings are in the range $[0.24, 0.26]$

Algorithm	Runs Near Optimum	Mean Best
CCEA	0	49.9994
cCCEA	233	143.1958
pCCEA	243	146.9373

Table 2: Comparison of CCEA, cCCEA, and pCCEA on MTQ_1 . We give the number of runs out of 250 which produce a near-optimal pair of variable settings, as well as the value of the highest-valued individual from each run, averaged across all 250 runs. Note cCCEA and pCCEA are roughly comparable, but outperform CCEA significantly.

Algorithm	Runs Near Optimum	Mean Best
cCCEA	177	142.6876
pCCEA	248	149.3980

Table 3: Control experiment comparing cCCEA and pCCEA on MTQ_2 . We give number of runs out of 250 which produce a near-optimal pair of variable settings, as well as the value of the highest-valued individual from each run, averaged across all 250 runs. Note cCCEA performs significantly worse on this problem than on MTQ_1 , whereas pCCEA performs comparably well.

simply because it has access to more collaboration information than CCEA. To address this question we ran a second experiment, applying both pCCEA and cCCEA to MTQ_2 . Recall that MTQ_2 is similar to MTQ_1 except the local optimum has objective value 125 rather than 50; thus, the spread between the global optimum and local optimum is lower. Table 3 gives the results for these two algorithms on MTQ_2 . In terms of the number of runs which produce a near-optimal pair, cCCEA does worse on MTQ_2 than on MTQ_1 . However, pCCEA performs comparably well on both MTQ_1 and MTQ_2 . The reason for this is the Pareto dominance mechanism permits the algorithm to see the value of an individual as soon as it lands on a peak, regardless of how high on the peak it falls.

5. DISCUSSION

We began with the question of modifying the CCEA to promote the discovery of global optima. pCCEA, a cooperative coevolutionary algorithm using a Pareto dominance mechanism to compare individuals, achieves this aim remarkably well. When compared with CCEA, pCCEA performs quite a bit better. When compared with another modification of CCEA which uses the same number of evaluations per generation (cCCEA), pCCEA performs comparably on MTQ_1 but significantly better on MTQ_2 . The explanation we give for the difference in performance on MTQ_2 is that cCCEA is sensitive to the relative, numerical objective values of the two peaks, whereas pCCEA is sensitive to informational differences between individuals and is insensitive to their numerical values. pCCEA's ability to find global optima suggests ideas from Pareto coevolution may fruitfully be applied to optimization with CCEAs in other domains.

In section 3 we observed that on MTQ_1 , informativeness and Pareto optimality coincide to some degree: certain Pareto optimal individuals are also differently informative.

This observation raises an intriguing question: could the relationship between informativeness and Pareto dominance yield a metric of how competitive or cooperative a domain is? It has been observed previously that in domains which have traditionally been called competitive, for instance game playing, Pareto dominance and informativeness are different: highly capable, dominant players are poor tests [2]. In the domain of chess playing, for example, Garry Kasparov is a dominant player, but the outcomes of a set of players' games against Kasparov would yield very little information about how the players compare to one another. In short, the discrepancy between informativeness and dominance in competitive domains is marked, whereas at least on these examples of cooperative domains, the two concepts appear related. In future work we intend to explore the relationship between these two orders and how their relationships characterize domains.

6. REFERENCES

- [1] A. Bucci and J. B. Pollack. Focusing versus intransitivity: Geometrical aspects of coevolution. In Erick Cantú-Paz et al., editor, *Genetic and Evolutionary Computation Conference - GECCO 2003*, volume 2723 of *Lecture Notes in Computer Science*, pages 250–261. Springer, 2003.
- [2] A. Bucci and J. B. Pollack. A Mathematical Framework for the Study of Coevolution. In K. De Jong, R. Poli, and J. Rowe, editors, *FOGA 7: Proceedings of the Foundations of Genetic Algorithms Workshop*, pages 221–235, San Francisco, CA, 2003. Morgan Kaufmann Publishers.
- [3] S. G. Ficici. *Solution Concepts in Coevolutionary Algorithms*. PhD thesis, Brandeis University, Waltham, Massachusetts, 2004.
- [4] S. G. Ficici and J. B. Pollack. Pareto Optimality in Coevolutionary Learning. In *European Conference on Artificial Life*, pages 316–325, 2001.
- [5] C. M. Fonseca and P. J. Fleming. An Overview of Evolutionary Algorithms in Multiobjective Optimization. *Evolutionary Computation*, 3(1):1–16, 1995.
- [6] J. Noble and R. A. Watson. Pareto coevolution: Using performance against coevolved opponents in a game as dimensions for Pareto selection. In L. Spector, E. Goodman, A. Wu, W. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshek, M. Garzon, and E. Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-2001*, pages 493–500, San Francisco, CA, 2001. Morgan Kaufmann Publishers.
- [7] L. Panait, R. P. Wiegand, and S. Luke. A sensitivity analysis of a cooperative coevolutionary algorithm biased for optimization. In Kalyanmoy Deb et al., editor, *Genetic and Evolutionary Computation Conference - GECCO 2004*, volume 3102 of *Lecture Notes in Computer Science*, pages 573–584. Springer, 2004.
- [8] M. A. Potter and K. A. De Jong. Cooperative coevolution: An architecture for evolving coadapted subcomponents. *Evolutionary Computation*, 8(1):1–29, 2000.
- [9] C. D. Rosin. *Coevolutionary Search Among Adversaries*. PhD thesis, University of California, San Diego, San Diego, California, 1997.
- [10] R. Watson and J. B. Pollack. Coevolutionary dynamics in a minimal substrate. In L. Spector et al., editor, *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-2001*, San Francisco, CA, 2001. Morgan Kaufmann Publishers.
- [11] R. P. Wiegand. *An Analysis of Cooperative Coevolutionary Algorithms*. PhD thesis, George Mason University, Fairfax, Virginia, 2003.